Universeller I/O-Erweiterungsbaustein EP30-PPIO64

Gunther Zielosko

1. Kurzbeschreibung

Mit dem neuen Erweiterungsbaustein EP30-PPIO64 steht dem Anwender ein Modul zur Verfügung, welches das Problem knapper I/Os mit freier Richtungswahl beim BASIC-Tiger® konsequent löst. Das Modul EP30-PPIO64 arbeitet wie eine PIO (PIO = Parallel Input Output) in anderen bekannten Mikroprozessor-Systemen, z.B. Z80-PIO. Es hat das gleiche Gehäuse wie z.B. das Erweiterungsmodul EP1-64HDE und beinhaltet insgesamt 64 I/O-Pins, die anders als bei den bisherigen Erweiterungsmodulen per Software in der Datenrichtung umschaltbar sind. Diese Pins können also als Eingänge oder Ausgänge konfiguriert werden. Die I/O-Anschlüsse sind portweise organisiert, die Ports 0 bis Port 5 sind 8-Bit-Ports, der Port 67 ist ein 16-Bit-Port.
Die Adressierung erfolgt ähnlich wie bei den bisherigen Erweiterungsmodulen durch einen 8-Bit-Adreß- und Datenbus (Bus-0...Bus-7) und einen zusätzlichen Adressen-Offset durch statische Adressen ADR4...ADR7. Es gibt die Steuerleitungen ACLK, DCLK und $\overline{\text{INE}}$.
Das PIO-Erweiterungsmodul EP30-PPIO64 arbeitet intern mit einer 3,3 V Versorgungsspannung, die im Modul selbst erzeugt wird. Am VCC-Pin wird, wie an jedem bisherigen Erweiterungsmodul nur eine Betriebsspannung von mindestens 5 V benötigt (Spannung des BASIC-Tiger-Systems). Im Falle des PIO-Erweiterungsmoduls EP30-UNI-I/O-01 werden wegen des eingebauten Reglers auch Spannungen bis 10 V am VCC-Pin toleriert. Eingangsspannungen dürfen zwischen 0 V (low) und 5 V (high) liegen, die Low- und High-Pegel der üblichen Logikfamilien (TTL, LS, HCT, HC und CMOS bei 5 V) werden sicher erkannt. Am Ausgang werden Pegel von 0 V (low) und 3,3 V (high) ausgegeben, was wiederum alle Standard-Logikfamilien erkennen.

2. Grundsätzliches zur Programmierung

Der PIO-Erweiterungsbaustein wird grundsätzlich nur mit dem XPort I/O – Erweiterungssystem (nur im Projektmodell PM_FULL) angesprochen und programmiert. Bei diesem XPort I/O – Erweiterungssystem kann man im Gegensatz zum traditionellen EPort I/O – Erweiterungssystem die Daten- und Steuerleitungen selbst wählen.
Vor der Benutzung des PIO-Erweiterungsbausteines EP30-UNI-I/O-01 muß eine Initialisierung des Xport I/O – Erweiterungssystems mit dem Befehl XSETUP vorgenommen werden, der den Daten- und Steuer-Port sowie die Steuerpins ACLK, DCLK und $\overline{\text{INE}}$ festlegt (siehe Handbuch Programmierung).
Es wird empfohlen, nicht die gleichnamigen Steuerpins des EPort I/O – Erweiterungssystems zu verwenden (Standardeinstellung nach Power On Reset ohne XSETUP), da es beim EPort-Erweiterungssystem infolge dessen unvollständiger Adreß-Dekodierung zu Konflikten mit anderen Komponenten (LC-Display, Tastatur usw.) kommen kann. Der PIO-Erweiterungsbaustein EP30-PPIO64 selbst hat eine vollständige Adreß-Dekodierung.

Alle Ein- und Ausgaben, Steuerworte und Zwischenspeicherungen werden über die Befehle XIN und XOUT vorgenommen.
Die einzelnen Ports werden grundsätzlich über die physikalischen Adressen, die der XIN- bzw. XOUT-Befehl über den BUS-Port benutzt, adressiert. Liegen die statischen Adressen des Bausteines A4 bis A7 auf logisch „0“, ergeben sich die Portadressen zu:

	Port-Adr.	
Port 0	00h	(ohne Offset, d.h. alle statischen Adressen auf logisch „0“)
Port 1	01h	
Port 2	02h	
Port 3	03h	
Port 4	04h	
Port 5	05h	
Port 6	06h	
Port 7	07h	

Mit den statischen Adressen des Bausteines A4 bis A7 können mit Jumpern oder DIL-Switches die Adressen jeweils um 16 angehoben werden, also:

00h → 10h → 20h → 30h → ... → F0h
↓ ↓ ↓ ↓
bei Belegung mit 1111b (A7, A6, A5, A4)
bei Belegung mit 0011b (A7, A6, A5, A4)
bei Belegung mit 0001b (A7, A6, A5, A4)
bei Belegung mit 0000b (A7, A6, A5, A4)

Mit dieser Adressierung über die statischen Adressen lassen sich weitere Bausteine anschließen oder es können Adressen-Konflikte mit bereits vorhandenen Bausteinen sicher vermieden werden. Wird z.B. mit A4 auf „1“ die Basis-Adresse auf 10h angehoben, bekommt dieser Baustein die Port-Adressen 10h ... 17h und die Steuer-Adressen 18h ... 1Fh.

Steueranweisungen, wie Richtungseinstellungen der Port-Pins oder Zwischenspeicher-Aktionen des 16-Bit-Ports 67 werden über Adressen gegeben, die um den Offset 8 höher im Modul-Adreßbereich liegen als die zu steuernden Ports. So werden z.B. alle Steuerbefehle, die den Port 0 auf der Adresse 00h betreffen, über die Adresse 08h ausgegeben:

	Port-Adr.	Steuer-Adr.	
Port 0	00h	08h	(ohne Offset, d.h. alle statischen Adressen auf logisch „0“)
Port 1	01h	09h	
Port 2	02h	0Ah	
Port 3	03h	0Bh	
Port 4	04h	0Ch	
Port 5	05h	0Dh	
Port 6	06h	0Eh	
Port 7	07h	0Fh	

3. Eigenschaften der I/O-Ports

Die Port-Pins sowie alle Steuerpins des Erweiterungsbausteines EP30-PPIO64 sind als Eingang konfiguriert 5V-tolerant, geben aber als Ausgang konfiguriert nur 3,3 V ab. Als Ausgang konfiguriert ist der entnehmbare Strom bei „Low“ und bei „High“ mehr als 20 mA, was deutlich über dem Angebot üblicher Logikfamilien liegt. Insgesamt muß aber die maximale Verlustleistung den gesamten Baustein beachtet werden! Bild 1 zeigt das Ausgangsverhalten (Pegel) in Abhängigkeit von der ohmschen Belastung (rot = High-Pegel bei Belastung gegen GND, blau = Low-Pegel bei Belastung gegen VCC +5V).

Als Eingänge konfigurierte Pins werden intern mit ca. 100kΩ bis 200kΩ gegen 3,3V (Pull-up) gezogen. Dieses Verhalten ist bei Beschaltung von Pins mit externen Leistungstreibern (z.B. FETs) insbesondere während der noch nicht abgeschlossenen Konfigurierung nach dem Power On Reset zu beachten, in dieser Phase sind alle I/O-Pins als Eingänge konfiguriert.

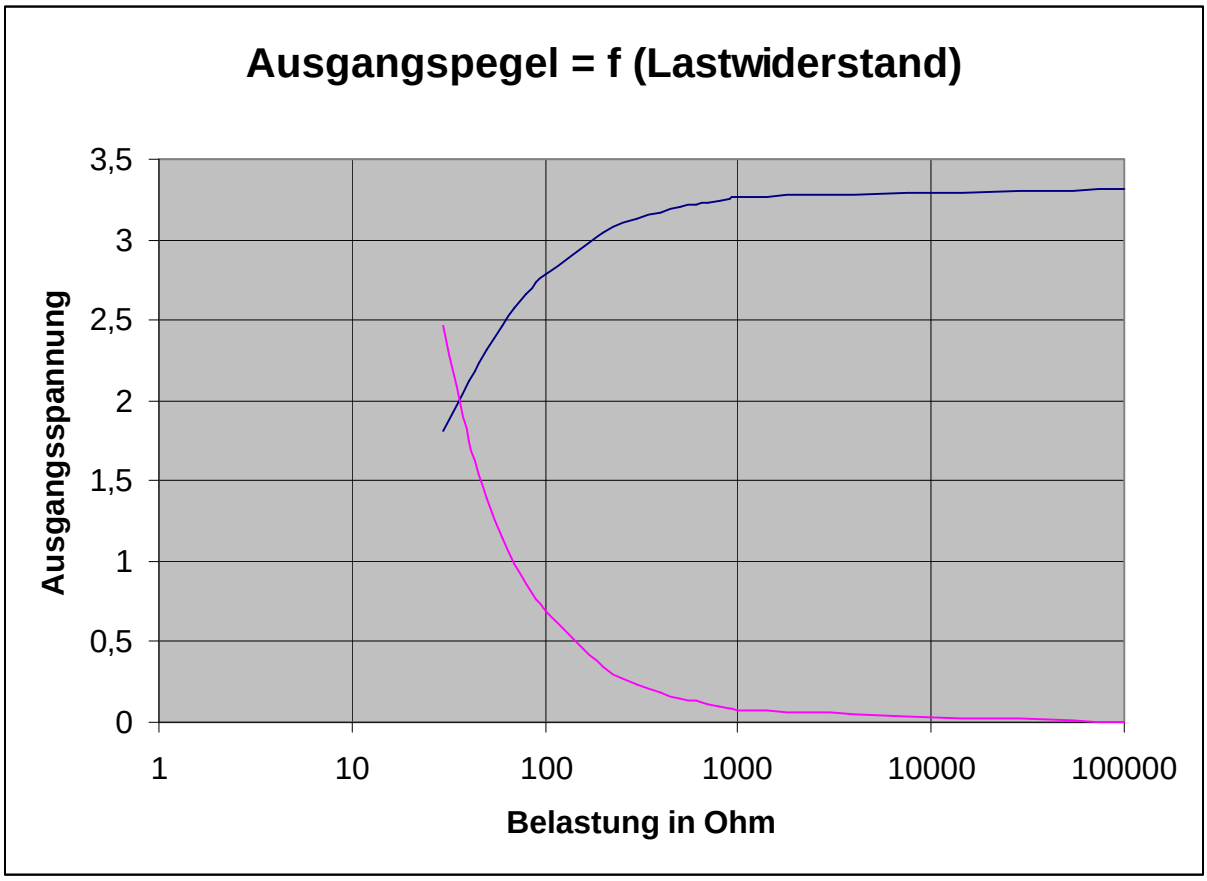


Bild 1 Verhalten der Ausgänge unter Last (blau = High-Pegel bei Belastung gegen GND, rot = Low-Pegel bei Belastung gegen VCC (+5V)

3.1. Ports 0 bis 5

Die Ports 0 bis 5 sind 8-Bit-Ports, deren Pins sowohl als Eingänge als auch als Ausgänge verwendet werden können. Die Umschaltung der Datenrichtung kann bitweise erfolgen, d.h. jedes Pin kann einzeln als Eingang oder Ausgang definiert werden.

Nach Power On Reset des PIO-Bausteines sind alle I/O-Pins als Eingänge konfiguriert. Die Festlegung der Datenrichtung eines Ports (z.B. Port 0 auf Adresse 00h) erfolgt mit einem Steuerbefehl XOUT, bei dem gesetzte Bits („1“) Eingänge und nicht gesetzte Bits („0“) Ausgänge symbolisieren (analog zum Befehl DIR_PIN). So wäre z.B. nach XOUT (08h, 0011 0001b):

Bit 0 von Port 0	Eingang
Bit 1 von Port 0	Ausgang
Bit 2 von Port 0	Ausgang
Bit 3 von Port 0	Ausgang
Bit 4 von Port 0	Eingang
Bit 5 von Port 0	Eingang
Bit 6 von Port 0	Ausgang
Bit 7 von Port 0	Ausgang

Ein XOUT-Befehl an Port 0 (an Adresse 00h gerichtet) legt die auszugebenden Daten nur an die als Ausgang definierten Pins, die als Eingang definierten Pins bleiben hochohmig. Die Ausgangsdaten werden solange gepuffert und stehen außen zur Verfügung, bis ein XOUT-Befehl etwas anderes festlegt.

Ein XIN-Befehl an Port 0 (an Adresse 00h gerichtet), holt die an den als Eingang konfigurierten Pins anliegende Information ab. Die als Ausgang definierten Pins werden dabei „rückgelesen“, d.h. es wird die Information gelesen, die vorher per XOUT-Befehl dort ausgegeben wurde. Der Auslesevorgang erfolgt nur einmal, es erfolgt keine Pufferung.

3.2. Port 67

Port 67 ist ein 16-Bit-Port mit besonderen Eigenschaften. Er dient speziell zur synchronen Eingabe oder Ausgabe von 16-Bit Worten. Dies ist z.B. sinnvoll für die Anwendung in Systemen mit DA-Wandlern mit mehr als 8 Bit Auflösung oder anderen Systemen, bei denen eine Übertragung von 8-Bit-Daten nacheinander zu Fehlinterpretationen führen kann. Aus diesem Grund macht eine bitweise Richtungseinstellung keinen Sinn, die Pins von Port 67 können entweder komplett als Eingänge oder komplett als Ausgänge definiert werden.
Diese Festlegung erfolgt mit dem Bit 0 des Steuerregisters auf Adresse 0Eh, eine „1“ konfiguriert den kompletten Port 67 als Eingangsport, eine „0“ als Ausgangsgangsport.

XOUT (0Eh,XXXXXX11b)	Port 67 hat 16 Eingänge
XOUT (0Eh,XXXXXX00b)	Port 67 hat 16 Ausgänge

Die Nutzung des 16-Bit-Ports 67 in einem 8-Bit-System, wie es der BASIC-Tiger® ist, erfordert bei der Programmierung Zwischenschritte zur Datenpufferung. Ein XOUT-Befehl kann nur 8 Bits ausgeben, ein XIN-Befehl nur 8 Bits einlesen. Deshalb müssen bei der Ausgabe zunächst die beiden Einzelregister 6 und 7 (die intern die Ports 6 und 7 mit ihren Adressen 06h und 07h darstellen) nacheinander mit Daten beschrieben werden, mit einem Steuerwort an die Steueradresse 0Eh erfolgt dann die gleichzeitige Ausgabe. Dabei gibt der Zustand von Bit 1 an, welche Funktion ausgeführt wird. Beim 0->1-Übergang werden die Daten aus einem „Schattenregister 67“ an alle Ausgänge übergeben, beim 1->0-Übergang werden die Daten aller Eingänge in ein „Schattenregister 67“ geschrieben. Das heißt, eine 16-Bit-Ausgabe muß in der Regel mit mehreren XOUT-Anweisungen nacheinander erfolgen.

XOUT (0Eh,XXXXXX00b) → Voraussetzung für Ausgabe! Port 67 ist Ausgang

XOUT (06h, 00110100b) → schreibt das Datenwort 00110100b in Registeradresse 06h
XOUT (07h, 01100111b) → schreibt das Datenwort 01100111b in Registeradresse 07h

XOUT (0Eh, XXXXXX00b)
XOUT (0Eh, XXXXXX10b) → mit L/H-Flanke am Bit 1 auf Adresse 0Eh werden die Daten der Registeradressen 06h und 07h an den Ausgängen des Port 67 gleichzeitig ausgegeben
XOUT (0Eh, XXXXXX00b)

Analog, allerdings umgekehrt funktioniert das Einlesen von 16-Bit-Daten:

XOUT (0Eh,XXXXXX11b) → Voraussetzung für Eingabe! Port 67 ist Eingang

XOUT (0Eh, XXXXXX11b)
XOUT (0Eh, XXXXXX01b) → mit H/L-Flanke am Bit 1 auf Adresse 0Eh werden die Daten der Eingänge des Ports 67 gleichzeitig in die Registeradressen 06h und 07h übernommen
XOUT (0Eh, XXXXXX11b)
L = XIN (06h) → der niederwertige Teil des 16-Bit-Wortes wird in Variable L eingelesen
M = XIN (07h) → der höherwertige Teil des 16-Bit-Wortes wird in Variable M eingelesen

Für das Ausgeben bzw. Einlesen von 16-Bit-Daten ist also immer ein kurzer Impuls am Bit 1 des Steuerwortes für Adresse 0Eh erforderlich.

4. Weitere Informationen

4.1. Pin „V3P“

Das Modul besitzt einen eigenen 3,3 V - Spannungsregler für den „inneren“ Bedarf. Der Pin V3P kann benutzt werden, um weitere externe Bausteine, die 3,3 V benötigen, zu betreiben. Unter Beachtung der Gesamtverlustleistung können hier beim Betrieb mit 5 V an Vcc des Bausteines problemlos bis zu 300 mA ohne Einschränkungen entnommen werden.

4.2. Pins „rsv.“

Diese Pins sind für Erweiterungen reserviert und müssen beim Erweiterungsmodul EP30-PPIO64 durch den Anwender unbedingt auf einen festen logischen Pegel (vorzugsweise GND, ansonsten +3,3 V oder + 5 V) gelegt werden werden. Der Pin 06b **muß** generell auf GND gelegt werden!

4.3. Signale ACLK, DCLK, $\overline{\text{INE}}$, Bus

Für die Kommunikation mit dem BASIC-Tiger-System wird das Bus-System Bus-0...Bus7 des Modules an den Port 6 (L60...L67) des BASIC-Tigers angeschlossen.
Als Steuerleitungen ACLK, DCLK und $\overline{\text{INE}}$ sollten nicht die gleichnamigen Pins des EPort-Systems, sondern unabhängige Pins eines anderen Ports genutzt werden. Hier bietet sich z.B. der seltener genutzte Port 7 an. Voraussetzung dafür ist, daß nicht andere Device-Treiber auf diesen zugreifen. Mit der Anweisung:

XSETUP (6, 7, 0, 1, 2, 4, 0) wird z.B. festgelegt

- Datenbus ist Port 6 des BASIC-Tigers
- Steuerbus 1 (CTRL1_PORT) ist Port 7 des BASIC-Tigers
- Als BIT_ACLK dient L70 des BASIC-Tigers (Low-aktiv, High/Low-Flanke)
- Als BIT_DCLK dient L71 des BASIC-Tigers (Low-aktiv, High/Low-Flanke)
- Als BIT_INE dient L72 des BASIC-Tigers (Low-aktiv, statisch)
- Steuerbus 2 (CTRL2_PORT) ist Port 4 des BASIC-Tigers *
- Als BIT_CE dient L40 des BASIC-Tigers *

* nicht benutzt, Eingabe trotzdem erforderlich!

Der PIO-Erweiterungsbaustein EP30-PPIO64 muß dann gemäß dieser Anweisung an das BASIC-Tiger-System angeschlossen werden.

5. Zusammenschaltung mit dem BASIC-Tiger

Das folgende Bild zeigt eine Variante der Beschaltung. Natürlich können für die Daten und Steuerpins im Rahmen der Möglichkeiten des XSETUP-Befehls auch andere Kombinationen gewählt werden. Für die Beispielbeschaltung ist das PIO-Testprogramm PIOX_01.TIG mit seinen Einstellungen gültig.

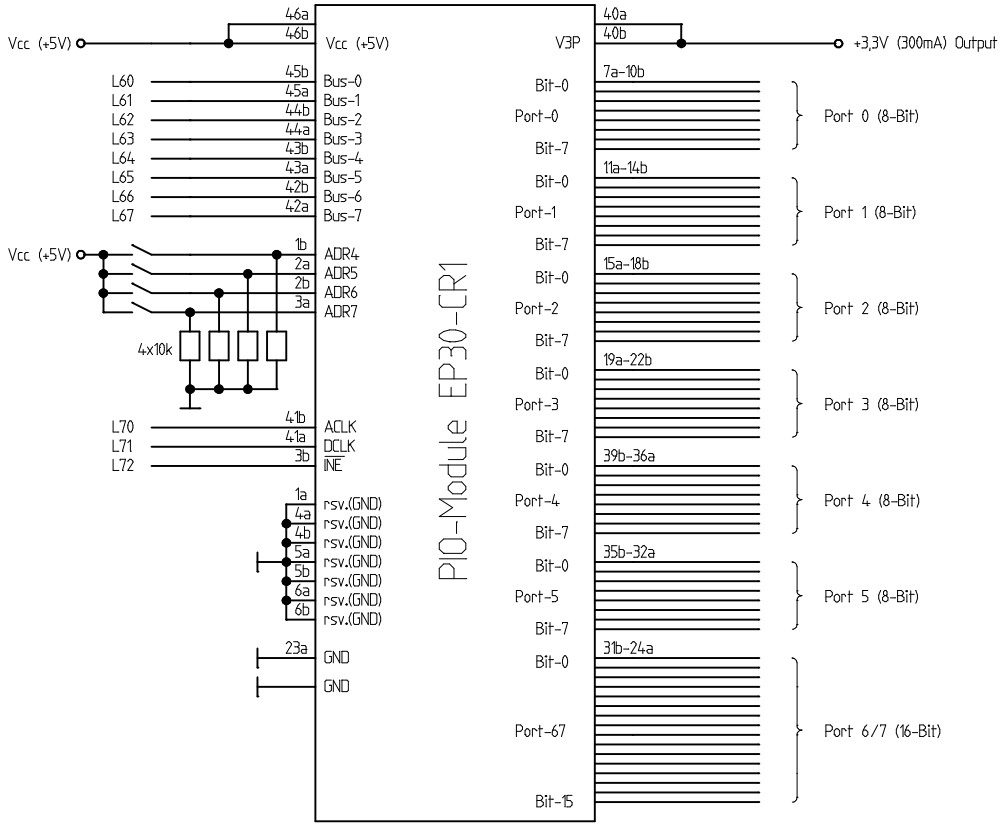
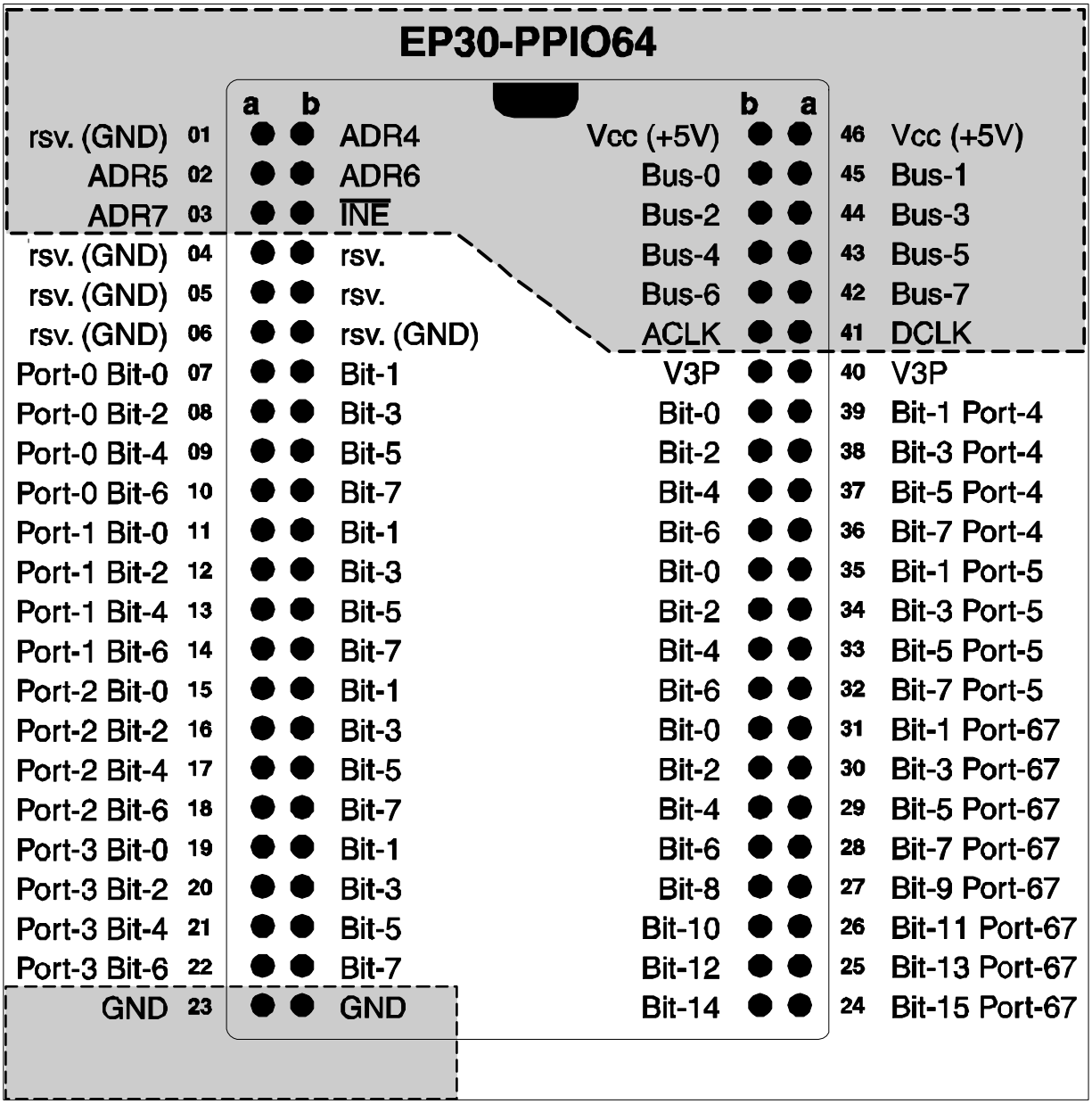


Bild 2 Beispiel für die Anschaltung des PIO-Moduls EP30-PPIO64 an den BASIC-Tiger

6. Pinbelegung



Bezeichnung	Funktion [1]	Bemerkung
GND	PWR	Ground
Vcc	PWR	Versorgungsspannung (+5V)
V3P	PWR	Ausgang des internen Reglers (+3,3V)
Bus-x	I/O	Systembus
DCLK	I	Data-Strobe (Low-aktiv, High/Low-Flanke)
ACLK	I	Adress-Strobe (Low-aktiv, High/Low-Flanke)
INE	I	Input-Enable (Low-aktiv, statisch)
ADRx	I	Moduladresse
Port-x Bit-y	I/O	Bit y des PIO-Ports x
Rsv.	I	Reserviert, Pins müssen an definierte logische Pegel geschaltet werden (vorzugsweise GND, + 3,3 V oder + 5 V! Pin 06b immer auf GND legen!

[1] PWR - Power; I - Input; O - Output; I/O - Input oder Output bzw. bidirektionaler Pin

7. Das Testprogramm PIOX_01.TIG

Ursprünglich zum Testen des Erweiterungsmoduls EP30-PPIO64 gedacht, demonstriert das Programm PIOX_01.TIG die neuen Möglichkeiten der programmierbaren I/Os. Die Basis des Programms ist, daß durch die freie Wahl von Aus- und Eingängen ein einfacher Selbsttest realisiert werden kann. Eine Hälfte der 64 I/Os arbeitet als Ausgang und sendet Daten, die andere Hälfte wird als Eingang konfiguriert und überprüft, ob die Ausgänge die erwartete Information schicken. Das Ganze ist als Zähler programmiert und macht somit wenig Aufwand, außerdem werden alle denkbaren Bitkombinationen erfaßt. Hat der Zähler den höchsten Wert (255) erreicht, werden Aus- und Eingänge vertauscht und ein zweiter Durchlauf beginnt. Zum Betrieb des Programms ist es lediglich erforderlich, die „linken“ Ports (also die Ports 0 – 3) mit den „rechten“ Ports (4, 5 und der 16-Bit-Port 67) bitweise zu verbinden. Am besten kann dies mit einem 32-poligen Flachkabel und Klemmsteckern erfolgen. Es kommt darauf an, daß Port 0/Bit 0 mit Port 4/Bit 0, Port 0/Bit1 mit Port 4/Bit 1 verbunden wird usw. Das Programm läuft bewußt so langsam ab, daß Sie auf dem Display des BASIC-Tigers den Fortschritt gut erkennen können. Im Übrigen erklärt sich das Programm über das Display selbst, so daß weitere Informationen zum Programm nicht erforderlich sind. Natürlich können Sie sich die Ausgangsinformationen mit einem Oszilloskop oder einer LED (Vorwiderstand 100 Ω) auch direkt ansehen.

Viel Spaß mit den „unbegrenzten Möglichkeiten“ der neuen programmierbaren I/Os.